THE 4D-MP GRAPHICS SUPERWORKSTATION: COMPUTING + GRAPHICS = 40 MIPS + 40 MFLOPS AND 100,000 LIGHTED POLYGONS PER SECOND

Forest Baskett, Tom Jermoluk, Doug Solomon

Silicon Graphics Computer Systems

Abstract

The 4D-MP graphics superworkstation brings 40 mips of computing performance to a graphics superworkstation. It also delivers 40 megaflops of geometry processing performance enabling 100,000 lighted, 4-sided, concave-tested polygons per second. This unprecedented level of computing and graphics processing in an office-environment workstation is made possible by the fastest available Risc microprocessors in a single shared memory multiprocessor design driving a tightly coupled, highly parallel graphics system. Aggregate sustained data rates of over one gigabyte per second are achieved by a hierarchy of buses in a balanced system designed to avoid bottlenecks.

Introduction

Graphics workstations from Silicon Graphics have always provided state of the art workstation computing performance tightly coupled with custom-silicon graphics subsystems that provided sufficient floating point performance to enable the real time display and movement of three dimensional objects. Progress in the underlying technology has contributed to not only faster and more powerful computing subsystems but also faster and more powerful graphics subsystems.

The computing subsystems have moved from one-half mips to two mips to ten mips to today's forty mips. These speed improvements have been accompanied by improvements in memory size, both primary and secondary, improvements in file system speed and capability, and improvements in compiling and operating system sophistication.

The graphics subsystems have progressed from 50,000 3-D vectors per second to 150,000 to today's 400,000 vectors per second. The functionality has also improved from less than 1000 flat shaded polygons per second to today's 100,000 Phong lighted, Gouraud shaded, four sided, concave tested polygons per second. Thus early systems could provide real time motion of moderately complex 3-D wire frames. Today's systems can provide real time motion of even more complex, remarkably realistic, lighted and shaded solid objects.

These advances have been made possible not only by advances in the speed and density of the underlying technology but also by our ability to design increasingly complex hardware and software systems in balanced configurations. Today's 4D-MP system has a hardware logic complexity of over two million gates, about the logic complexity of the last generation of commercial mainframe computers.

Computing System Architecture

The key new element of the 4D-MP workstation design is its use of a tightly coupled multiprocessor. The particular configuration we describe in this paper is a 4 processor system, although the architecture will support from one to sixteen processors. The key design philosophy of the system is a hierarchy of buses, each tailored to the functional needs for which it is intended.

Figure 1 is a block diagram of the major components of the multiprocessor computing section of the system. Several of the important buses of the system are shown in this diagram. The sync bus is the bus that provides high speed synchronization between the main processors of the system in support of fine grained parallelism. The processor buses allow full speed access to instructions and data from the individual first level caches of each main processor. The read and write buffers allow for the efficient flow of information between the processors and the main memory of the system. The second level data cache provides the additional bandwidth needed to support an automatically consistent shared memory computing model with such high speed processors. The MP bus then supports the protocols for consistent data sharing and the high speed block transfers for fast data sharing among the processors, the main memory, the I/O subsystem, and the graphics subsystem.

The Sync Bus. The sync bus is designed for the synchronization needs of a multiprocessor supporting efficient fine grained parallelism. The goal is for a single application to be able to make efficient use of parallel processors even at the individual loop level, in addition to the kinds of larger grained parallelism found in many system simulation applications and the even larger grained parallelism found in the process structure of most Unix systems.

The sync bus provides 65,000 individual test-and-set variables. These variables are in a special part of the physical address space. They are addressed as memory and can be allocated to individual applications by the operating system. They are arranged 64 to a page and can be mapped into the virual address space of an application. The operating system itself makes use of them to provide very fine grained locks for the control variables of the operating system. The operating system is thus a very parallel, fully symmetric multiprocessing operating system. In other words, the Silicon Graphics verion of Unix V.3 is a well developed parallel processing application on the 4D-MP and its speed demonstrates some of the power of this approach to high speed computing.

Because the sync bus can provide synchronization operations to applications with an overhead of only tens of cycles, many programming and compiler techniques developed for vector processors are also suitable for this kind of parallel processor. For example, strip mining, the technique of taking a long vector and breaking it into a number of strips for use by a vector register, can be used by taking a long vector and breaking it into 4 strips, one for each processor. This technique, when properly applied, can result in super-linear speedups of applications because of the potential for taking advantage of the private cache organization of the multiprocessor.

The sync bus also provides for the distribution of interrupts from one processor to another, or from the I/O system to appropriate processors. The flexibility of the interrupt distribution system means that the operating system can provide scheduling algorithms that support the power of private caches rather than disrupt it.

The processor bus. Each processor provides both an address bus and a data bus that can support sustained data transfers at 8 bytes every clock cycle. Thus this four processor system has a total processor-to-cache sustainable bandwidth of 512 megabytes per second. In addition, the organization of this bandwidth on a modular basis means that smallar and larger configurations are possible in such a fashion that the bandwidth shrinks or grows with the processing power. This kind of bandwidth modularity is not only sensible, it is economical.

The modular processor bandwidth and private caches are also what make super-linear speedups possible. A super-linear speedup is defined as running an application more than N times faster on an N processor system than on a 1 processor system. Since large problems are often constrained by data bandwidth needs as much as by compute cycle needs, the addition of more data bandwidth in conjunction with more processor cycles can result in super-linear speedups. This phenomenon may not be common but its existence helps demonstrate the value and power of this type of system organization.

The first level caches, separate for instructions and data, are all 64 kilobytes. The total cache size is 512 kilobytes in a 4 processor system. The instruction caches are fed by a read

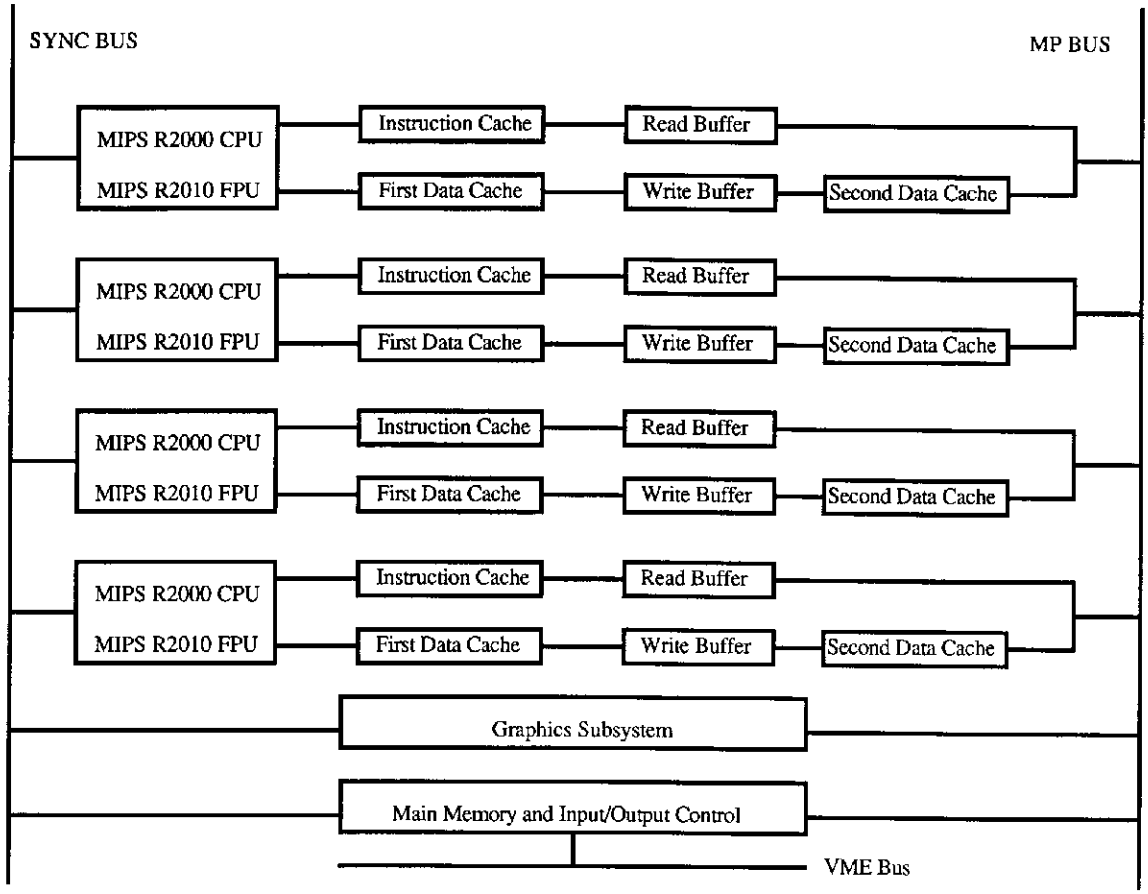

Figure 1. Computing System Architecture

buffer and the data caches drive a write buffer. These buffers also provide a convenient point for an asynchronous interface between the processors and the MP Bus. This asynchronous interface provides an added degree of modularity in the overall system. The clock speed of all the processors and the MP Bus is 16 megahertz but we anticipate faster processors in the future. The asynchronous interface makes it possible to upgrade to faster processors without having to replace the entire system.

Each second level data cache is 64 kilobytes in size, organized as 4 thousand lines of 16 bytes each. This cache provides the block transfer capability that the MP bus supports and it also provides the additional bandwidth for the cache line tag storage necessary to keep all the individual caches in a consistent state. The second level data cache watches every transaction on the MP bus and checks for transactions involving data in its data storage. This checking is performed by matching every address on the MP bus with the addresses in the tag storage section of the second level data cache. The first level data cache is always a subset of the second level data cache so consistent data is guaranteed. In addition, since all the caches are physical address caches rather than virtual address caches, there are no synonym problems caused by mapping different virtual addresses to the same physical address. The difficult system level issues that arrise when dealing with multiple virtual address caches are not present in this system.

The MP Bus. The MP bus is a pipelined, block transfer bus that supports the cache consistency protocol as well as providing 64 megabytes of sustained data bandwidth between the processors, the memory and I/O system, and the graphics subsystem. Because the sync bus provides for efficient synchronization between processors, the cache consistency protocol is designed to support efficient data sharing between processors. If a cache consistency protocol has to support synchronization as well as sharing, a compromise in the efficiency of the data sharing protocol may be necessary to improve the efficiency of the synchronization operations. With separate buses for these separate functions, each bus can be designed to support its function without compromises for the other function.

The cache consistency protocol used is sometimes called the Illinois protocol. Each second level data cache maintains state values for each cache line. A line can be in one of four states: invalid, private read, shared read, and private write. If a processor writes into a shared read line, the processor must first invalidate other copies of that cache line before the write can be completed. Simultaneous writes into a shared read line by several processors will result in write misses in all the processors except the one which successfully accquired the MP bus and issued an invalidate operation on it. In addition, any synchronization operation performed by a processor on the sync bus must not complete until all pending write activity by that processor is complete.

With these simple rules enforced by the hardware protocols of the sync bus and the MP bus, efficient synchronization and efficient data sharing are achieved in a simple shared memory model of parallel processing. The only data not visible to any processor is the data in another processor's registers. That sort of invisible data is handled by the usual safeguards in modern optimizing compilers.

The physical structure of the MP bus is 32 address lines and 64 data lines. An MP bus transaction is six cycles in length although the last two data transfer cycles can overlap with the first two cycles of the next transaction, resulting in a sustainable data transfer rate of 64 megabytes per second in a very economical configuration. New data arrives in the last two cycles. Old data from a swapped cache line is carried in the middle two cycles. Addresses are transferred in the first two cycles. Bus arbitration is pipelined and doesn't add to the cycle cost of transactions.

How does 64 megabytes of MP bus bandwidth balance with the computing requirements of this 4 processor configuration? Each individual MIPS R2000 processor and R2010 coprocessorat 16 megahertz would be a 12 mips processor in the absence of the cost of coordination overhead in a multiprocessor configuration. A conservative rating of the individual processors is thus about 10 mips. If we use the rule of thumb that we need about one megabyte per second of memory bandwidth for each mip, we need 40 megabytes per second for a 4 processor configuration. Each coprocessor has a peak floating point rate of 8 megaflops, double precision, and we can deliver 1.6 megaflops on applications like Linpack. If we use the rule of thumb that we need about 8 megabytes per second per delivered megaflop, we need about 50 megabytes per second to deliver over 6 megaflops of double precision floating point performance. Thus the MP bus bandwidth can easily support the computing needs of the 4 processor configuration and still have bandwidth left for the graphics subsystem. These are rough rules of thumb, of course, but they have been found to be useful in a variety of computing systems and they give some indication of the balance of this system. Benchmarks of real applications are the true measure of a system's performance, though.

Graphics System Architecture

The graphics system of the 4D-MP is a new version of the 4D GT graphics system introduced by Silicon Graphics in 1987. The architectural structure this graphics system is not changed but improved bandwidth to the graphics system has resulted in improved overall delivered graphics performance. Figure 2 is a diagram of the major components of this graphics system.

Silicon Graphics graphics systems have always provided special purpose floating point engines, called geometry engines, to accomodate the floating point computational needs for 3 dimensional computer graphics. The GT graphics system continues that practice at a remarkable level of performance. The five geometry engines in the geometry pipeline of the graphics system each have a peak single precision floating point computational rate of 20 megaflops, for a total peak rate of 100 megaflops. Peak rates are never achieved, even in special purpose configurations like the geometry engines but we observe actual rates of 40 megaflops when displaying complex images. Thus we achieve about 40 percent efficiency with the geometry pipeline.

It has recently become fashionable in some quarters to imagine that the floating point computational needs for geometry processing could be supplied by a general purpose floating point vector unit. Since vector units rarely deliver more than 15 to 20 percent of their peak rates for actual computations, it would presumably require a vector unit with a peak rate of 200 single precision megaflops to equal the realized geometry processing power of this graphics system. In addition, bus

bandwidth requirements would go up substantially if our delivered megaflops versus bus bandwidth rule of thumb has any merit. Again, only actual application benchmarks will tell for sure.

The geometry pipeline feeds a polygon processor designed to handle arbitrarily shaped polygons. This generality usually simplifies applications and also makes it possible to handle the most common case of four sided polygons at a rate almost equal to our rate for three sided polygons. The more common situation is for the four sided polygon rate to be less than half the rate for triangles in those systems designed around triangle primitives. Thus we can render 100,000 independent four sided polygons that are Phong lighted, Gouraud shaded, concave tested, clip tested, and Z-buffered. For triangles, we can render 120,000.

The polygon processor then feeds a set of edge processors that iterate down the edges of a polygon to determine a sequence of vertical spans that must be rendered. These spans are broadcast to a set of 5 span processors over the pixel bus. Each of the span processors is responsible for one fifth of the spans and picks up every fifth span to render.

Each span processor broadcasts the pixel addresses of each pixel in the span to a set of four image engines. Each image engine is responsible for one twentieth of the pixels and picks up every fourth pixel from its span processor. This two dimensional interleaving of pixel processing means that even small polygons can be rendered with twenty way parallelism.

The resulting fill rates for pixels are 80 million pixels per second without Z buffering and 40 million pixels per second with Z buffering.

Of course, vector drawing is a special case and takes place at the rate of 400,000 vectors per second for plain vectors or 200,000 vectors per second for anti-aliased vectors. All these numbers are for polygons and vectors that are 10 pixels on a side.

Summary

The 4D-MP graphics superworkstation uses both general purpose and special purpose multiprocessing to provide minisupercomputer performance in a workstation package with the highest performance real time 3-D graphics available. The use of a hierarchy of buses, each designed to economically provide efficient performance for their particular functions is the organizing principle of this system. The resulting package illustrates the deliverable power of parallel processing.

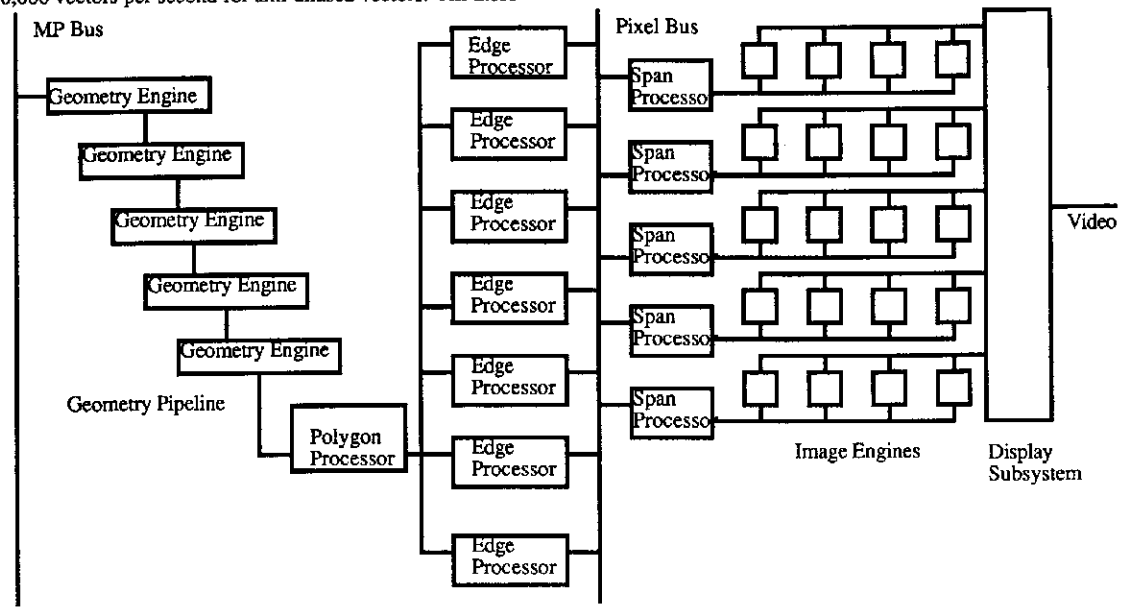

Figure 2. Graphics System Architecture